CLAIMS

1. A process for converting binary words from a non-encoded format to a compressed encoded format, in which, in said compressed encoded format, binary words are, at least in part, represented by encoded bit sequences that are shorter than the respective binary word in the non-encoded format, said encoded bit sequences being selected according to a statistical recurrence of the respective words in the non-encoded format, and in which associated with binary words with higher recurrence are encoded bit sequences including numbers of bits that are accordingly smaller, a correspondence between binary words in non-encoded format and the encoded bit sequences associated therewith being established by indices of at least one encoding vocabulary, the process comprising:

arranging said indices in an orderly sequence;

organizing said sequence of indices in groups of vectors;

splitting each group of vectors into a given number of vectors;

as well as at least one of the following operations:

encoding said vectors independently of one another; and

calculating and storing in a data structure, for each group of vectors, a starting address of a compressed block, or else differences, expressed in bytes, with respect to a last complete address appearing in said data structure.

2. The process according to claim 1, applied to binary words making up instructions of a processor which has a row cache of given size, , wherein the organizing step includes forming said groups of vectors with size equal to the row cache of the processor, so that each vector contains a number of indices equal to the ratio between the size of said cache row and the number of vectors contained in each group.

3. The process according to claim 2, wherein both the number of vectors comprised in each group and the number of indices comprised in each vector are powers of two.

4. The process according to claim 1, further comprising splitting said binary words in said non-encoded format into a given number of subwords, said binary subwords being converted into said encoded format by resorting to respective encoding vocabularies.

5. The process according to claim 4, wherein splitting said binary words includes splitting said binary words into two binary subwords, one of which corresponds to a high part and the other to a low part of the binary word in the non-encoded format.

6. The process according to claim 1, further comprising constituting said at least one encoding vocabulary, reserving to binary words of low recurrence respective encoded bit sequences which comprise a number of bits that may be not less than the number of bits comprised in the corresponding binary word in non-encoded format.

7. The process according to claim 1, further comprising storing said groups of vectors, and storing, for each of said groups of vectors, a respective starting address.

8. The process according to claim 7, wherein storing a respective starting address includes providing an address-translation table for saving the starting addresses of said groups of vectors.

9. The process according to claim 1, further comprising determining, for each of said vectors, a corresponding norm resulting from the addition of the indices contained in the vector itself.

10. The process according to claim 9, further comprising subjecting said norm to variable-length coding.

11. The process according to claim 10, wherein said norm is encoded with a bit string comprising a tag field and a field of bits representing a real value of the norm, the norm values with higher recurrence being represented by tag fields and by fields representing the real value, which are shorter than the tag fields and the fields representing the real value reserved to the norm values with lower statistical recurrence.

12. The process according to claim 9, wherein, in order to convert said binary words from said non-encoded format into said encoded format, the least significant bits of each non-encoded binary word are attributed to the corresponding encoded bit sequence, proceeding with a series of operations of division by two corresponding to shift operations of bits comprised in the binary word in non-encoded format, said shift operations being performed until the corresponding norm reaches a predetermined value.

13. The process according to claim 12, wherein said encoded bit sequence is organized as a set of fields comprising:

a field identifying the number of shifts made;
identification of the bits subjected to shifting;
the resulting norm value; and
a shortest encoded bit sequence.

14. The process according to claim 13, wherein organizing the encoded bit sequence as a set of fields includes locating said field identifying the bits subjected to shifting immediately after the corresponding shortest encoded bit sequence.

15. The process according to claim 13 wherein, in order to convert a given binary word from said encoded format into said non-encoded format, the process comprises the operations of:

identifying the corresponding group of vectors in the encoded format;

reading, starting from said group of vectors, said field identifying how many shifts have been made;

reading the identification of the bits subjected to shifting;

reading the resulting norm value;

reading said shortest encoded bit sequence; and

constructing said binary word in said non-encoded format starting from the information read.

16. The process according to claim 9, further comprising:

defining a number of successive indices that can be encoded together as a single encoded word;

building a look-up table, of size LxK, where L is said number of successive indices that can be encoded together and K represents a maximum norm possible for the encoded words, then obtaining the encoded word starting from L successive indices according to the relation

$$encoded = \sum_{i=1}^{L} \sum_{j=1}^{x} Ntable[i, j]$$

where "encoded" represents the encoded word, the summations extend for i ranging from 1 to L and for j ranging from 1 to x_i, respectively, and Ntable[i, j] represents a set of integer-coordinate points that identify a pyramid of i-dimensions with norm j, said norm

expressing a sum of the absolute values of the co-ordinates of the points that make up the pyramid.

17. The process according to claim 16,wherein, in order to convert a given binary word from said encoded format into said non-encoded format, the process comprises the operations of:

reading said given binary word and a corresponding norm value;locating a matrix value in a last row of said table in a column identified by the norm value;

comparing the matrix value with the given binary word in said encoded format;

if the given binary word is smaller than the matrix value read, subtracting the given binary word from the matrix value read and reducing the norm value by one until the comparison is satisfied; and

if the given binary word is greater than the matrix value read, a difference between the starting norm and the effective norm is the last of the encoded indices, and the process is restarted.

18. The process according to claim 17, wherein the comparing step includes:

comparing the given binary word directly with a number contained in a row indicated by the corresponding norm value; and

calculating a resulting index to be used for decoding, on the basis of the winning position within the table thus identified.

19. The process according to claim 18, wherein, to obtain the indices to be used, coding proceeds by obtaining a last index as equal to the difference between the remaining part of the norm value and the value of the norm value after all the subtractions have been made on them, obtaining instead the penultimate index as residual input value.

20. The process according to claim 16, further comprising:

forming a table which gathers all the encoded words possible for said norm value in ascending order, so that also the original words that form the encoded word are ordered with respect to one another;

determining, starting from a certain encoded word, the corresponding binary word in non-encoded format on the basis of the difference between the maximum norm available and the real norm of the encoded word, defining an index in the first table of the group where the input word is located; and

determining to which group the current encoded word belongs, by counting how many non-negative differences exist for the position identified by said index, which identifies the current subvector, and the number of encoded words that belong to each preceding group.

21. The process according to claim 1 wherein said encoded bit sequences map, within them, an address identifying the corresponding binary word in non-encoded format, so that each of said encoded bit sequences can be converted into the corresponding binary word in non-encoded format independently of other encoded bit sequences.

22. A processor system comprising:

plural cache memories;

a memory;

a peripheral unit;

a bus connected to the memory and peripheral unit;

a bus interface;

a processing unit configured to interact via the bus interface and the bus with the memory and peripheral unit; and

a converter modulelocated in at least one intermediate position between:

one of said cache memories and said bus; and

said bus and at least one of said memories, the converter module including means for converting binary words from a non-encoded format to a compressed encoded format, in which, in said compressed encoded format, binary words are, at least in part, represented by encoded bit sequences that are shorter than the respective binary word in the non-encoded format, said encoded bit sequences being selected according to a statistical recurrence of the respective words in the non-encoded format, and in which associated with binary words with higher recurrence are encoded bit sequences including numbers of bits that are accordingly smaller, a correspondence between binary words in non-encoded format and the encoded bit sequences associated therewith being established by indices of an encoding vocabulary, the means including:

means for arranging said indices in an orderly sequence;

means for organizing said sequence of indices in groups of vectors;

means for splitting each group of vectors into a given number of vectors; and

means for performing at least one of the following operations:

encoding said vectors independently of one another; and

calculating and storing in a data structure, for each group of vectors, a starting address of a compressed block, or else differences, expressed in bytes, with respect to a last complete address appearing in said data structure.

23. The system according to claim 22, wherein said plurality of caches includes an instruction cache and said converter module is set between the instruction cache and said bus interface.

24. The system according to claim 22 wherein said converter module is set between said bus and a program memory associated with said processing unit.

25. The system according to claim 22, wherein the plurality of cache memories includes a cache organized on two levels and said converter module is set between said two cache levels.

26. A computer program product directly loadable in a memory of a digital processor and comprising software code portions for performing a process for converting binary words from a non-encoded format to a compressed encoded format when said product is run on the processor, in which, in said compressed encoded format, binary words are, at least in part, represented by encoded bit sequences that are shorter than the respective binary word in the non-encoded format, said encoded bit sequences being selected according to a statistical recurrence of the respective words in the non-encoded format, and in which associated with binary words with higher recurrence are encoded bit sequences including numbers of bits that are accordingly smaller, a correspondence between binary words in non-encoded format and the encoded bit sequences associated therewith being established by indices of at least one encoding vocabulary, the process comprising:

arranging said indices in an orderly sequence;

organizing said sequence of indices in groups of vectors;

splitting each group of vectors into a given number of vectors;

as well as at least one of the following operations:

encoding said vectors independently of one another; and

calculating and storing in a data structure, for each group of vectors, a starting address of a compressed block, or else differences, expressed in bytes, with respect to a last complete address appearing in said data structure.